Agilent Technologies

Advanced Design System 2011.01

Feburary 2011
Layout Vs Schematic

Acknowledgments

version 2.1 of the License, or (at your option) any later version. This library is distributed in the hope that it will be useful,but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details. You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA. unRAR copyright: The decompression engine for RAR archives was developed using source code of unRAR program.All copyrights to original unRAR code are owned by Alexander Roshal. unRAR License: The unRAR sources cannot be used to re-create the RAR compression algorithm, which is proprietary. Distribution of modified unRAR sources in separate form or as a part of other software is permitted, provided that it is clearly stated in the documentation and source comments that the code may not be used to develop a RAR (WinRAR) compatible archiver. 7-zip Availability: http://www.7-zip.org/

AMD Version 2.2 - AMD Notice: The AMD code was modified. Used by permission. AMD copyright: AMD Version 2.2, Copyright © 2007 by Timothy A. Davis, Patrick R. Amestoy, and Iain S. Duff. All Rights Reserved. AMD License: Your use or distribution of AMD or any modified version of AMD implies that you agree to this License. This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version. This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details. You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA Permission is hereby granted to use or copy this program under the terms of the GNU LGPL, provided that the Copyright, this License, and the Availability of the original version is retained on all copies.User documentation of any code that uses this code or any modified version of this code must cite the Copyright, this License, the Availability note, and "Used by permission." Permission to modify the code and to distribute modified code is granted, provided the Copyright, this License, and the Availability note are retained, and a notice that the code was modified is included. AMD Availability: http://www.cise.ufl.edu/research/sparse/amd

UMFPACK 5.0.2 - UMFPACK Notice: The UMFPACK code was modified. Used by permission. UMFPACK Copyright: UMFPACK Copyright © 1995-2006 by Timothy A. Davis. All Rights Reserved. UMFPACK License: Your use or distribution of UMFPACK or any modified version of UMFPACK implies that you agree to this License. This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version. This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details. You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA Permission is hereby granted to use or copy this program under the terms of the GNU LGPL, provided that the Copyright, this License, and the Availability of the original version is retained on all copies. User documentation of any code that uses this code or any modified version of this code must cite the Copyright, this License, the Availability note, and "Used by permission." Permission to modify the code and to distribute modified code is granted, provided the Copyright, this License, and the Availability note are retained, and a notice that the code was modified is included. UMFPACK Availability: http://www.cise.ufl.edu/research/sparse/umfpack UMFPACK (including versions 2.2.1 and earlier, in FORTRAN) is available at http://www.cise.ufl.edu/research/sparse . MA38 is available in the Harwell Subroutine Library. This version of UMFPACK includes a modified form of COLAMD Version 2.0, originally released on Jan. 31, 2000, also available at http://www.cise.ufl.edu/research/sparse . COLAMD V2.0 is also incorporated as a built-in function in MATLAB version 6.1, by The MathWorks, Inc. http://www.mathworks.com .

COLAMD V1.0 appears as a column-preordering in SuperLU (SuperLU is available at http://www.netlib.org). UMFPACK v4.0 is a built-in routine in MATLAB 6.5. UMFPACK v4.3 is a built-in routine in MATLAB 7.1.

Qt Version 4.6.3 - Qt Notice: The Qt code was modified. Used by permission. Qt copyright: Qt Version 4.6.3, Copyright (c) 2010 by Nokia Corporation. All Rights Reserved. Qt License: Your use or distribution of Qt or any modified version of Qt implies that you agree to this License. This library is free software; you can redistribute it and/or modify it under the
terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version. This library is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details. You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA Permission is hereby granted to use or copy this program under the terms of the GNU LGPL, provided that the Copyright, this License, and the Availability of the original version is retained on all copies.User
documentation of any code that uses this code or any modified version of this code must cite the Copyright, this License, the Availability note, and "Used by permission." Permission to modify the code and to distribute modified code is granted, provided the Copyright, this License, and the Availability note are retained, and a notice that the code was modified is included. Qt Availability: http://www.qtsoftware.com/downloads Patches Applied to Qt can be found in the installation at: $HPEESOF_DIR/prod/licenses/thirdparty/qt/patches. You may also contact Brian Buchanan at Agilent Inc. at brian_buchanan@agilent.com for more information.

The HiSIM_HV source code, and all copyrights, trade secrets or other intellectual property rights in and to the source code, is owned by Hiroshima University and/or STARC.

Errata The ADS product may contain references to "HP" or "HPEESOF" such as in file names and directory names. The business entity formerly known as "HP EEsof" is now part of Agilent Technologies and is known as "Agilent EEsof". To avoid broken functionality and to maintain backward compatibility for our customers, we did not change all the names and labels that contain "HP" or "HPEESOF" references.

Warranty The material contained in this document is provided "as is", and is subject to being changed, without notice, in future editions. Further, to the maximum extent permitted by applicable law, Agilent disclaims all warranties, either express or implied, with regard to this documentation and any information contained herein, including but not limited to the implied warranties of merchantability and fitness for a particular purpose. Agilent shall not be liable for errors or for incidental or consequential damages in connection with the furnishing, use, or performance of this document or of any information contained herein. Should Agilent and the user have a separate written agreement with warranty terms covering the material in this document that conflict with these terms, the warranty terms in the separate agreement shall control.

Technology Licenses The hardware and/or software described in this document are furnished under a license and may be used or copied only in accordance with the terms of such license. Portions of this product include the SystemC software licensed under Open Source terms, which are available for download at http://systemc.org/ . This software is redistributed by Agilent. The Contributors of the SystemC software provide this software "as is" and offer no warranty of any kind, express or implied, including without limitation warranties or conditions or title and non-infringement, and implied warranties or conditions merchantability and fitness for a particular purpose. Contributors shall not be liable for any damages of any kind including without limitation direct, indirect, special, incidental and consequential damages, such as lost profits. Any provisions that differ from this disclaimer are offered by Agilent only.

About ADS Desktop Layout Versus Schematic

ADS Desktop LVS is a basic **Layout Versus Schematic** check that validates designs prior to submitting them to the foundry. It is a tool that checks whether a schematic and a layout are in synchronization or not. The check is done for the ADS database elements such as PDK components and sub-designs.

You can use ADS Desktop LVS to:

- Detect component, nodal, and parameter mismatches.
- Component-count information which makes easy to detect missing components.
- Analysis based on database comparison and do not require a rule deck.
- Can be run on any ADS design containing components (PDK library elements, hierarchical components or custom models).
- Designs do not need to be synchronized with ADS Layout/Schematic Design Synchronization.

> **Note**
> Appropriate Design Kits should be available for the check to be performed completely.

Key Features

- Selecting an error highlights the corresponding elements in schematic and layout.
- Supports a full hierarchical check.
- Can be used in combination with Design Synchronization but does not depend on it.
- Does not require instance names to be equivalent, but will potentially use this information to resolve ambiguities (especially for Ports).
- Can compare a layout and schematic from different designs.
- Can be run from either schematic or layout.
- Provides a component-count breakdown for foundry submission.

Assumptions

- Consistent port numbering in schematic and layout (P1 in schematic maps to P1 in layout).
- Consistent hierarchy in both schematic and layout.
- Layout devices are connected with traces or transmission lines (not paths or polygons).

Recommendations

- Fix missing components first. This will reduce the number of nodal mismatches.
- If you see a message “LVS could not determine n components. Fix Nodal errors(s) to resolve this”. Then proceed to fix the nodal errors. In that process you can determine the missing components too.
- Use the component matching information in the Component tab to highlight the equivalent components.
- Use the keyboard right arrow to expand the browser. Use the keyboard left arrow to contract.
- Analyzing component parameters may require expressions to be evaluation and can take several seconds. You can chose to disable the parameter check when running ADS Desktop LVS.

Flattened Artwork in Layout

Isolate the layout artwork in a sub-component. To do this:

1. Select the layout artwork and click **Edit > Component > Create Hierarchy**.
2. Open the *Schematic* window for the custom component and enter the corresponding schematic.
3. Place an instance of the custom component in the schematic.
   The database traversal will stop when it encounters a design with parameterized or fixed artwork. ADS Desktop LVS will check to make sure the custom component exists in both schematic and layout, and that it is connected to the same pins.

Schematics Containing Terms and GNDs

S-Parameter simulation requires terminals connected to grounds whereas a layout will have Ports. To avoid having these appear as component mismatches, it is recommended that the schematic be pushed down one level of hierarchy. The schematic and layout will then have an equivalent number of Ports. To do this, select the circuit elements and click **Edit > Component > Create Hierarchy**.

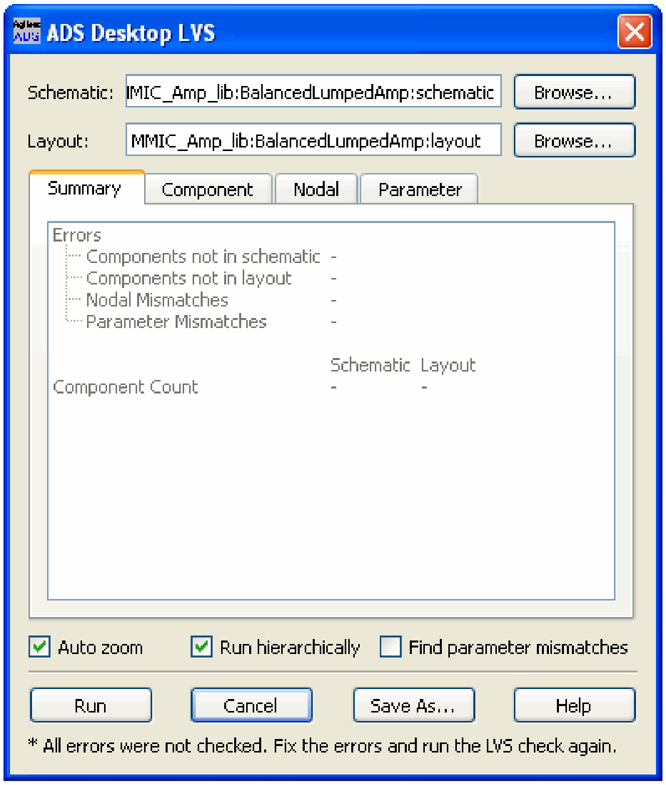
Recommended Uses

- Layout Verification prior to Foundry Submission
  - Use ADS Desktop LVS to validate that the schematic and layout are equivalent.
  - Run with parameter check enabled to verify parameters are consistent (number of turns on a spiral inductor).
  - Use the keyboard right arrow to expand the browser. Use the keyboard left arrow to contract.
  - Layout Verification during Design Creation.
  - Use ADS Desktop LVS throughout the design cycle to find and fix errors early.
  - Use the component-count information to quickly identify if a different number of components exist in schematic and layout. Correct these errors first before addressing nodal mismatches.
  - Use the component-count information to inspect the location of components. Use the keyboard down arrow to step through each component type. The instances will highlight in the schematic and layout.
- Layout Verification for Team Design
  - Use ADS Desktop LVS to check for consistency between a schematic and layout created by different designers or a layout created manually.
  - Use ADS Desktop LVS to check for completeness during the integration phase (combining different parts of a design into one schematic and corresponding layout).
- Layout Verification after Design Generation
  - ADS has the ability to auto-generate a layout from a schematic (and visa versa). (See Layout>Generate/Update Layout...). If you then edit either the schematic or layout you can use ADS Desktop LVS to check for component, nodal or parameter mismatches.
- Parameter Check
  - Analyzing component parameters may require expressions to be evaluated and can take several seconds. You can chose to disable the parameter check when running ADS Desktop LVS.
- Schematic with Ports
  - S-Parameter simulations require terminals connected to grounds whereas a layout will have Ports. To avoid having these appear as component mismatches, it is recommended that the schematic be pushed down one level of hierarchy. The schematic and layout will then have the equivalent number of Ports. To do this, select the circuit elements and execute the menu pick Edit>Component>Create Hierarchy.

Running ADS Desktop LVS Check

To run ADS Desktop LVS check:

1. Start ADS and open an existing workspace or create a new workspace.
2. Create a layout or open an existing layout.
3. In **Layout** window, select **Tools** > **ADS Desktop LVS** or you can select **Tools** > **ADS Desktop LVS** in **Schematic** window. The *ADS Desktop LVS* dialog opens.



Schematic: IMIC_Amp_lib:BalancedLumpedAmp:schematic

Layout: MMIC_Amp_lib:BalancedLumpedAmp:layout

Errors
- Components not in schematic -
- Components not in layout -
- Nodal Mismatches -
- Parameter Mismatches -

	Schematic	Layout
Component Count	-	-

☑ Auto zoom ☑ Run hierarchically □ Find parameter mismatches

* All errors were not checked. Fix the errors and run the LVS check again.

4. Click **Browse** to select a Schematic from the *Open Design* dialog box.
5. Choose **Run hierarchically** to run the LVS check from current layout to the lowest level.
6. Click **Summary** tab to view the summary of the LVS run.
7. Click **Component** tab to display the component errors.
8. Click **Nodal** tab to display the nodal errors.
9. Click **Parameter** tab to display the parameter errors.
10. Choose **Auto Zoom** to zoom to the problematic instance in the schematic or layout window.
11. Click **Run** to run the LVS tool.
12. Click **Cancel** to close the LVS dialog box.
13. Click **Save As...** to save LVS report to a text file.

Types of LVS Checks

The following are the types of LVS checks that can be done:

- Component Count
- Component Mismatch
- Nodal Mismatch
- Parameter Mismatch

Component Count

All schematic and layout components that contains pins are considered for counting. If Run Hierarchically option is used, counting is done hierarchically from the top level schematic and layout till the lowest level. A detailed hierarchical report of Component

Count can be found in the detailed text report (**ADS Desktop LVS** > **Save As**).

Component Mismatch

All schematic and layout components that contains pins are considered for comparison. The tool checks for a one-to-one correspondence between all the schematic and layout components. The check is performed irrespective of component's instance names. The check is based on the component type and the adjacent connections to other components.

Nodal Mismatch

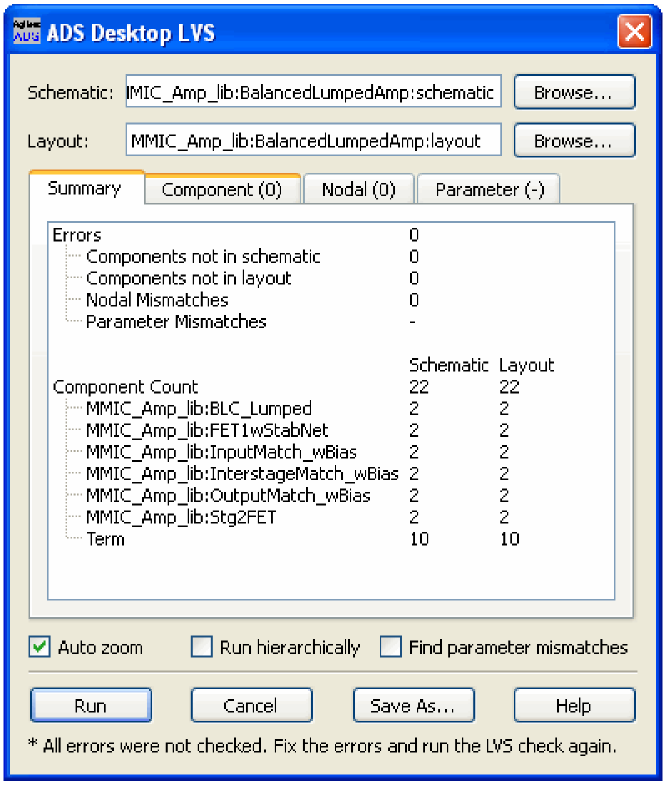
Nodal Mismatch tab lists schematic and layout nodal connection problems. The error list shows the last possible matching component beyond which there exists a nodal mismatch in schematic or in layout. Nodal tab shows errors in *<schematic component name> <instance name> / <layout component name> <instance name>* format.

Parameter Mismatch

Parameter Mismatch check is done when the 'Find parameter mismatches' option is selected.

An Example of Running ADS Desktop LVS

1. Open a workspace. For example, open _$HPEESOF_DIR/examples/MW_Ckts/MMIC_Amp_wrk.
2. Open a layout. For example, in this case the layoutn is *BalancedLumpedAmp*.
3. Go to **Tools > ADS Desktop LVS** and click **Run**. You should see the below LVS run data.



Schematic: IMIC_Amp_lib:BalancedLumpedAmp:schematic

Layout: MMIC_Amp_lib:BalancedLumpedAmp:layout

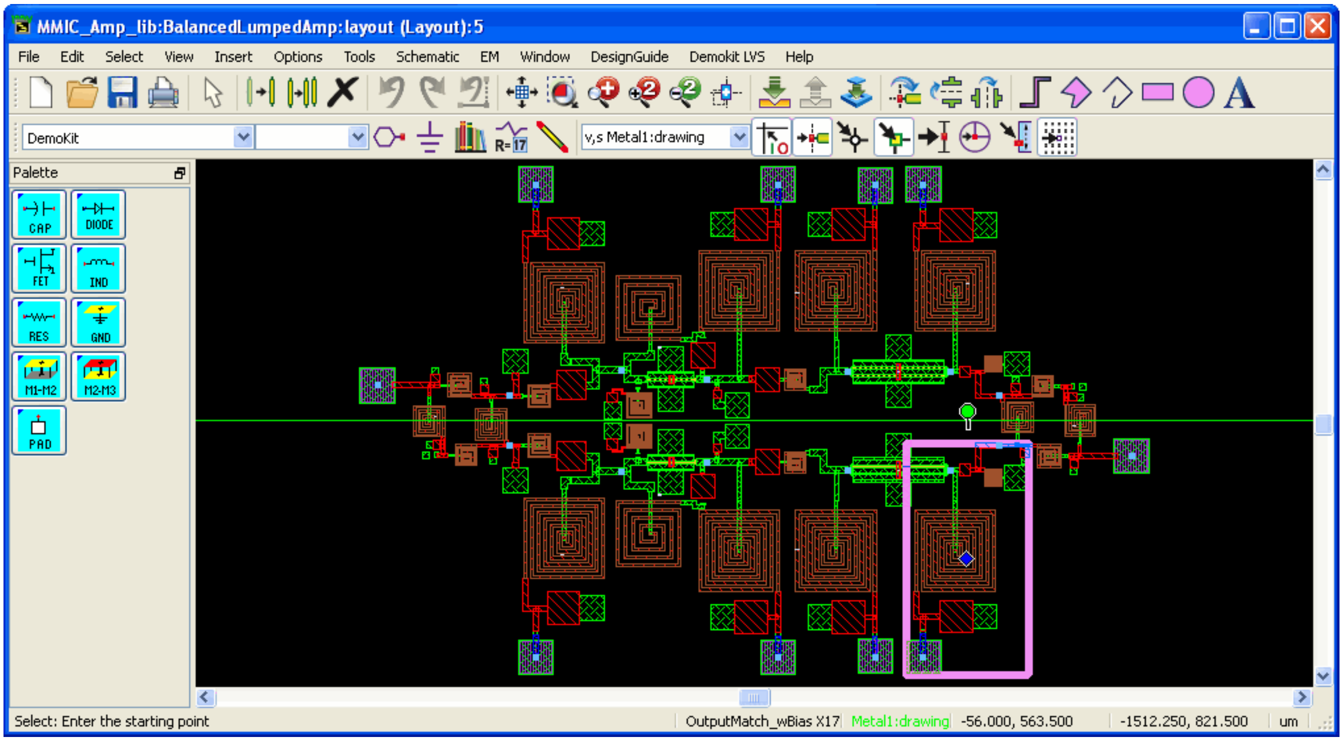
```
Errors                                  0
   Components not in schematic          0
   Components not in layout             0
   Nodal Mismatches                     0
   Parameter Mismatches                 -

                                        Schematic  Layout
Component Count                         22         22
   MMIC_Amp_lib:BLC_Lumped              2          2
   MMIC_Amp_lib:FET1wStabNet            2          2
   MMIC_Amp_lib:InputMatch_wBias        2          2
   MMIC_Amp_lib:InterstageMatch_wBias   2          2
   MMIC_Amp_lib:OutputMatch_wBias       2          2
   MMIC_Amp_lib:Stg2FET                 2          2
   Term                                 10         10
```

☑ Auto zoom ☐ Run hierarchically ☐ Find parameter mismatches

* All errors were not checked. Fix the errors and run the LVS check again.

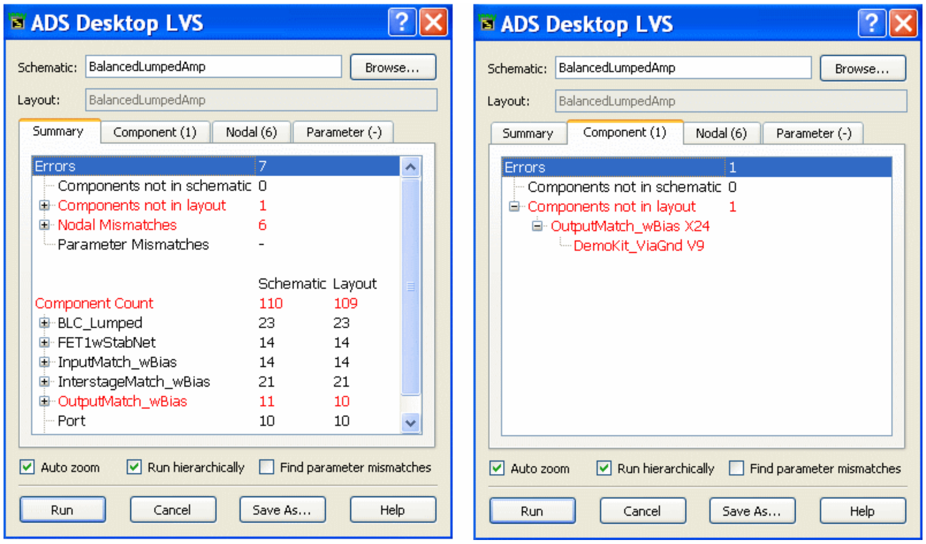
4. Now on BalancedLumpedAmp (layout), select instance X24 (OutputMatch_wBias).
   - Now click **View > Push Into Hierarchy**.
   - This opens up another instance of design, From this design, delete instance V9 (DemoKit_ViaGnd).
   - Click **View > Pop Out of Hierarchy**.

- 
MMIC_Amp_lib:BalancedLumpedAmp:layout (Layout):5
File Edit Select View Insert Options Tools Schematic EM Window DesignGuide Demokit LVS Help
DemoKit
v,s Metal1:drawing
Palette
CAP
DIODE
FET
IND
RES
GND
M1-M2
M2-M3
PAD

Select: Enter the starting point | OutputMatch_wBias X17 | Metal1:drawing | -56.000, 563.500 | -1512.250, 821.500 | um

> **Note**
> The LVS check is always done for the current layout that is shown on the layout window and the schematic that is shown on the ADS Desktop LVS dialog box.

5. Now on ADS Desktop LVS dialog, select **Run Hierarchically** and click **Run**. After running the check, you can click to select the tabs, *Summary* and *Component* to view the data as shown in the figure below:



ADS Desktop LVS

Schematic: BalancedLumpedAmp
Layout: BalancedLumpedAmp

Summary | Component (1) | Nodal (6) | Parameter (-)

Errors	7	
Components not in schematic	0	
Components not in layout	1	
Nodal Mismatches	6	
Parameter Mismatches	-	
	Schematic	Layout
Component Count	110	109
BLC_Lumped	23	23
FET1wStabNet	14	14
InputMatch_wBias	14	14
InterstageMatch_wBias	21	21
OutputMatch_wBias	11	10
Port	10	10

☑ Auto zoom ☑ Run hierarchically □ Find parameter mismatches

Run | Cancel | Save As... | Help

ADS Desktop LVS

Schematic: BalancedLumpedAmp
Layout: BalancedLumpedAmp

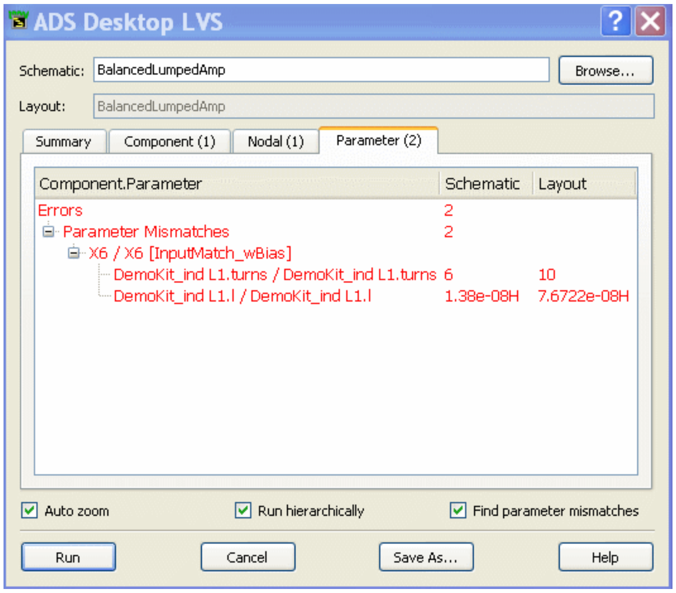
Summary | Component (1) | Nodal (6) | Parameter (-)

Errors	1
Components not in schematic	0
Components not in layout	1
OutputMatch_wBias X24	
DemoKit_ViaGnd V9	

☑ Auto zoom ☑ Run hierarchically □ Find parameter mismatches

Run | Cancel | Save As... | Help

6. Select instance X6 (InputMatch_wBias) from *BalancedLumpedAmp* layout, and click **View** > **Push Into Hierarchy**.
7. From the instance L1 (DemoKit_ind), modify the value of *turns* parameter from 6 to 10.
8. Click **View** > **Pop Out of Hierarchy**.
9. Make sure you select the **Find parameter mismatches** option. Click **Run** from the *ADS Desktop LVS* dialog box. The following data is displayed in the Parameter section.



Schematic: BalancedLumpedAmp

Layout: BalancedLumpedAmp

Component.Parameter	Schematic	Layout
Errors	2	
Parameter Mismatches	2	
X6 / X6 [InputMatch_wBias]		
DemoKit_ind L1.turns / DemoKit_ind L1.turns	6	10
DemoKit_ind L1.l / DemoKit_ind L1.l	1.38e-08H	7.6722e-08H

☑ Auto zoom ☑ Run hierarchically ☑ Find parameter mismatches

LVS Report

Click **Save As...** to generate LVS report. The following is an example of a LVS report:

```
Advanced Design System 2009 Update 1
Copyright(c) Agilent Technologies 1989-2009

Schematic: C:\users\default\MMIC_Amp_prj\networks\BalancedLumpedAmp
Layout: C:\users\default\MMIC_Amp_prj\networks\BalancedLumpedAmp
Component Comparison: Number of Errors: 0
Component Count: Schematic 22 Layout 22
 BLC_Lumped                 2          2
 FET1wStabNet               2          2
 InputMatch_wBias           2          2
 InterstageMatch_wBias      2          2
 OutputMatch_wBias          2          2
 Port                      10          10
 Stg2FET                    2          2
Parameter Comparison: Number of Errors: 0

Schematic: C:\users\default\MMIC_Amp_prj\networks\InputMatch_wBias
Layout: C:\users\default\MMIC_Amp_prj\networks\InputMatch_wBias
Component Comparison: Number of Errors: 0
Component Count: Schematic 14 Layout 14
 DemoKit_M1_M2        1          1
 DemoKit_M2_M3        1          1
 DemoKit_ViaGnd       2          2
 DemoKit_bondPad      1          1
 DemoKit_cap          3          3
 DemoKit_ind          2          2
 DemoKit_res          1          1
 Port                 3          3
Parameter Comparison: Number of Errors: 2
 DemoKit_ind L1: parameter turns has different values: "6 " in schematic "10 " in layout.
 DemoKit_ind L1: parameter l has different values: "1.38E-008 H" in schematic "7.6722E-008 H" in layout.


Schematic: C:\users\default\MMIC_Amp_prj\networks\OutputMatch_wBias
Layout: C:\users\default\MMIC_Amp_prj\networks\OutputMatch_wBias
Component Comparison: Number of Errors: 2
Not in Schematic: 0
Not in Layout:
 V9[DemoKit_ViaGnd]
Nodal discrepancy in Schematic: 0
Nodal discrepancy in Layout:
 C20[DemoKit_cap] corresponding Schematic equivalent C20[DemoKit_cap]
Component Count: Schematic 11 Layout 10
 DemoKit_M2_M3        1          1
 DemoKit_ViaGnd       2          1
 DemoKit_bondPad      1          1
 DemoKit_cap          2          2
 DemoKit_ind          2          2
 Port                 3          3
Parameter Comparison: Number of Errors: 0
```

About ADS Desktop Layout Versus Schematic

ADS Desktop LVS is a basic **Layout Versus Schematic** check that validates designs prior to submitting them to the foundry. It is a tool that checks whether a schematic and a layout are in synchronization or not. The check is done for the ADS database elements such as PDK components and sub-designs.

You can use ADS Desktop LVS to:

- Detect component, nodal, and parameter mismatches.
- Component-count information which makes easy to detect missing components.
- Analysis based on database comparison and do not require a rule deck.
- Can be run on any ADS design containing components (PDK library elements, hierarchical components or custom models).
- Designs do not need to be synchronized with ADS Layout/Schematic Design Synchronization.

Note
Appropriate Design Kits should be available for the check to be performed completely.

Key Features

- Selecting an error highlights the corresponding elements in schematic and layout.
- Supports a full hierarchical check.
- Can be used in combination with Design Synchronization but does not depend on it.
- Does not require instance names to be equivalent, but will potentially use this information to resolve ambiguities (especially for Ports).
- Can compare a layout and schematic from different designs.
- Can be run from either schematic or layout.
- Provides a component-count breakdown for foundry submission.

Assumptions

- Consistent port numbering in schematic and layout (P1 in schematic maps to P1 in layout).
- Consistent hierarchy in both schematic and layout.
- Layout devices are connected with traces or transmission lines (not paths or polygons).

Recommendations

- Fix missing components first. This will reduce the number of nodal mismatches.
- If you see a message “LVS could not determine n components. Fix Nodal errors(s) to resolve this”. Then proceed to fix the nodal errors. In that process you can determine the missing components too.
- Use the component matching information in the Component tab to highlight the equivalent components.
- Use the keyboard right arrow to expand the browser. Use the keyboard left arrow to contract.
- Analyzing component parameters may require expressions to be evaluation and can take several seconds. You can chose to disable the parameter check when running ADS Desktop LVS.

Using Calibre LVS Link

Calibre LVS lets you perform Layout vs. Schematic check for your designs against the LVS rules. When you choose Calibre LVS, ADS exports your layout design to GDSII and your schematic design to SPICE (Calibre specific) netlist file. It also generates a Calibre LVS runset file.

ADS feeds the following files as inputs to the Calibre LVS engine:

- GDSII layout
- SPICE netlist
- LVS rule file
- Runset file

When LVS runs on Calibre, the results are displayed in the Calibre RVE error viewer. Calibre RVE highlights the errors on the ADS layout as well as on the ADS schematic.

ADS-Calibre LVS Link runs in two modes:

- Local
- Remote

Local mode is applicable if ADS and Calibre are available on the same Linux or Solaris machine and *Remote* if you are running ADS on a Windows, Linux, or Solaris machine and have network access to Calibre on another Linux or Solaris machine. Do not confuse this *Local* and *Remote* mode with the Local and Remote mode present in Calibre itself. The terminology *Local* and *Remote* are from ADS perspective, throughout this document.

Prerequisites For Running ADS-Calibre LVS Link

The following are prerequisites for running ADS-Calibre LVS link:

1. UNIX account on the machine on which Calibre is installed
2. Configuring secure shell or ssh on the local and remote machine. For any user, it requires one time setup to configure ssh. ssh configuration requires generation of a private-public key pair (See SSH Configuration)
3. Mounting the home directory
4. XServer software
5. CNEX definitions of individual component type present in the schematic design.

Each of the above requirements is described in detail in the following sections.

UNIX Account

You must have a UNIX (Linux/Solaris) account on the machine on which Calibre is installed. The account MUST have the *PATH*, *HOME* and other relevant UNIX environment variables appropriately set. The Calibre installation path must be specified as part of the *PATH* environment variable.

SSH Configuration

Generating a Private-Public Key Pair in Windows

Note
SSH Private-Public key generation is not required if you are running ADS-Calibre LVS link in *Local* mode.

1. Click **Start > Run** from your Windows task bar and enter `cmd` to open a DOS command prompt.

2. Enter `echo %HOME%` to check your home directory.
   Ensure that the home directory is the same as the home directory set in ADS. ADS home directory can be confirmed by running the following command:
   `de_info(getsysenv("HOME"));` in ADS command line tool (**Tools > Command Line**).
3. Enter `cd %HOME%` to go to home directory.
4. Create the directory `.ssh` in the home directory by entering
   `mkdir .ssh`
5. Generate a private-public key pair by entering
   `ssh-keygen -t dsa -f id_dsa -P ''`. If you get the message for `ssh-keygen not found` then include `%HPEESOF_DIR/tools/bin` in your *PATH* environment variable.
   This should result in two files, id_dsa (private key) and id_dsa.pub (public key).

Generating a Private-Public Key Pair in Solaris/Linux

Note
SSH Private-Public key generation is not required if you are running ADS-Calibre LVS link in *Local* mode.

1. Enter `echo $HOME` to check your home directory.
   Ensure that the home directory is the same as the home directory set in ADS. ADS home directory can be confirmed by running the command
   `de_info(getsysenv("HOME"));` in ADS command line tool (**Tools > Command Line**).
   Enter `cd $HOME` to go to home directory.
2. Create the directory `.ssh` in the home directory by entering
   `mkdir .ssh`
3. Generate a private-public key pair by entering
   `ssh-keygen -t dsa -f $HOME/.ssh/id_dsa -P ''`. This should result in two files, *id_dsa* (private key) and *id_dsa.pub* (public key).

Copying the Public Key to the Remote Machine

Note
These steps are not required if you are running ADS-Calibre LVS link in *Local* mode.

To copy the public key to the remote machine, perform the following steps:

1. Copy file *id_dsa.pub* from the local machine to the remote machine.
2. On the remote machine run the following commands:

   ```
   $ cat id_dsa.pub >> $HOME/.ssh/authorized_keys
   $ chmod 0600 $HOME/.ssh/authorized_keys
   ```

   Depending on the version of OpenSSH, the file authorized_keys2 may also be required:

   ```
   $ cat id_dsa.pub >> $HOME/.ssh/authorized_keys2
   $ chmod 0600 $HOME/.ssh/authorized_keys2
   ```

   An alternative is to create a link from authorized_keys to authorized_keys2:

   ```
   $ cd $HOME/.ssh
   $ ln -s authorized_keys authorized_keys2
   ```

   On the local (client) LINUX machine test the results by connecting to the remote (server) machine:

   ```
   $ ssh -i $HOME/.ssh/id_dsa -l <remote_user_name> <remote_machine_name>
   ```

   If using PC as the client machine, test the results using the following set of

commands
Open a Windows cmd prompt window and cd to your ADS HOME directory

```
C:\Documents and Settings\users1> cd C:\users\default
C:\users\default> <ads_install_dir>\tools\bin\ssh -i .ssh/id_dsa -l <remote_user_name>
<remote_machine_name>
```

You must login for the first time from a shell or DOS command prompt. You may be asked for confirmation yes/no. Enter `yes` to proceed. If everything is correct, you can login without any password to the remote machine. In case of any problem, consult your system administrator.

DRC Mapped Drive

> **Note**
> DRC Mapped Drive is not required if you are running Calibre LVS in *Local* mode.

1. Declare `DRC_MAPPED_DRIVE` environment variable.
2. Set `DRC_MAPPED_DRIVE` to the home directory of your remote UNIX machine.
3. The home directory must be mounted on your local machine (the machine on which you are running ADS).
   For example, assume that you are running ADS on your Windows machine and Calibre is installed on a UNIX machine "machine1.wlv.xyz.com". If your home directory on "machine1.wlv.xyz.com" is */home/adsuser*, then you must mount that home directory onto your Windows drive, say Z: (or any other valid drive) and set the value of `DRC_MAPPED_DRIVE` to ***Z:***
   On the other hand, if you are running ADS on a Linux/Solaris machine and Calibre is installed on another Linux/Solaris machine, then set the value of `DRC_MAPPED_DRIVE` to */home/adsuser*. It is assumed that you home directory is NFS mounted, so that it is accessible from all Linux/Solaris machines on your LAN. ADS creates a job folder inside the location specified by `DRC_MAPPED_DRIVE`. The job folder contains all the necessary files and folders required to run Calibre LVS.

XServer Software

If you are running ADS from a Windows machine, you must run a XServer software before running Calibre LVS. The XServer software allows exporting the display of all UNIX/LINUX GUI windows (in the present case the Calibre GUI windows) to the Windows machine. There are numerous XServer software available, many of which are freeware and some are licensed. Popular ones are Reflection-X, Xming, XWin32, Exceed, Cygwin etc.

CNEX SPICE Definitions

For each of the component types in the schematic design, there is a corresponding CNEX SPICE definition. For ADS built-in components, these CNEX definitions already exist. For components pulled from a PDK, the CNEX definition should be created.

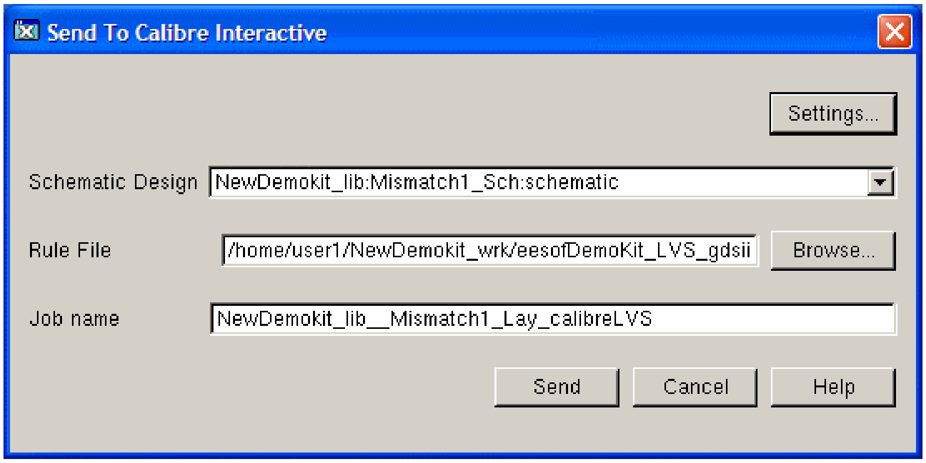
ADS has a utility for creating the CNEX definition. It is available in the Schematic window. Click on **Tools > Netlist Export > Edit Component Definition**. The wizard helps you in creating the CNEX definition. Select **calibre** as the tool. Refer ADS documentation for help in creating CNEX definition.

Running ADS-Calibre LVS Link in Local Mode

> **Note**
> You can run only one LVS Job at a time.

To prepare your Linux or Solaris machine:

1. Declare the `MGLS_LICENSE_FILE` environment variable.
2. Declare the `MGC_HOME` environment variable.
3. Open your Layout design as well as the corresponding Schematic designs. You should open only one window for a given Layout design
and another window for the Schematic design.
4. From your Layout window, click **Tools > Calibre LVS > Send To Calibre Interactive**.



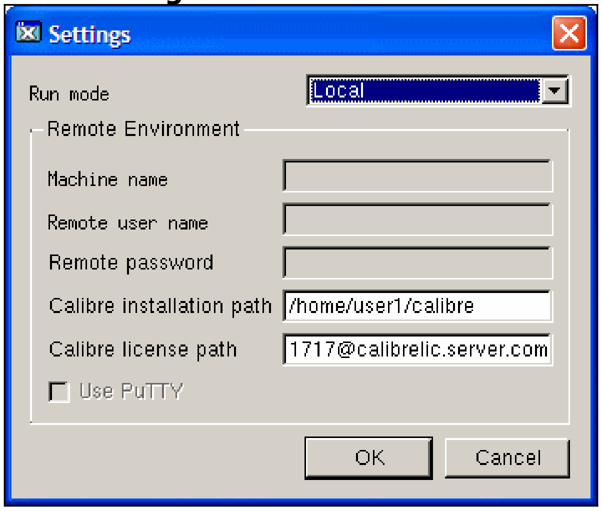
Send To Calibre Interactive

Schematic Design: NewDemokit_lib:Mismatch1_Sch:schematic

Rule File: /home/user1/NewDemokit_wrk/eesofDemoKit_LVS_gdsii

Job name: NewDemokit_lib__Mismatch1_Lay_calibreLVS

5. Click **Settings**.



Settings

Run mode: Local

Remote Environment

Machine name:

Remote user name:

Remote password:

Calibre installation path: /home/user1/calibre

Calibre license path: 1717@calibrelic.server.com

□ Use PuTTY

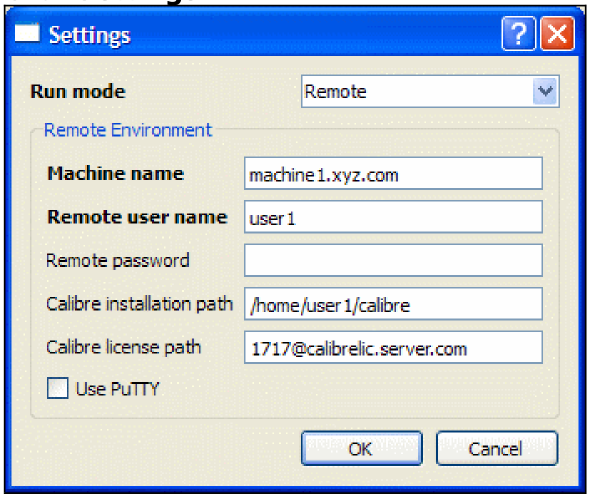
6. Choose **Local** for the *Run mode* in the *Settings* dialog.
7. As an alternative to declaring the `MGLS_LICENSE_FILE` variable, you can specify the Calibre license path in the dialog shown below.
8. You can also specify the Calibre installation path in the dialog instead of declaring the `MGC_HOME` variable.
9. Click **OK** in the *Settings* dialog to return to the *Send To Calibre Interactive* dialog.
10. In the *Schematic Design* drop down box, specify the schematic design against which you would like the Layout design to be checked for
LVS errors.
11. Click **Browse** to choose the *Calibre LVS Rule file*.
12. The Job name is automatically filled up by ADS. You can edit this name.
13. Click **Send** to start the LVS flow.

Running ADS-Calibre LVS Link in Remote Mode

> **Note**
> You can run only one LVS Job at a time.

In remote mode, Calibre LVS is run on a remote machine using a secure shell. To prepare your remote machine:

1. Declare the `MGLS_LICENSE_FILE` environment variable on the local machine from which you are running ADS. Its value should be same as that is defined in the remote machine UNIX account settings. As an alternative to declaring the `MGLS_LICENSE_FILE` variable, you can specify the Calibre license path in the dialog shown below.
2. Declare the `MGC_HOME` environment variable on the local machine from which you are running ADS. Its value should be same as that is defined in the remote machine UNIX account settings. As an alternative to declaring the `MGC_HOME` variable, you can specify the Calibre installation path in the dialog shown below.
3. Declare the `DRC_MAPPED_DRIVE` environment variable.
4. Ensure that secure shell, ssh is available and configured in the local and remote machine.
5. From your Layout window, click **Tools > Calibre LVS > Send To Calibre Interactive**.
6. Click **Settings**.



Settings

Run mode: Remote

Remote Environment

Machine name: machine1.xyz.com

Remote user name: user1

Remote password:

Calibre installation path: /home/user1/calibre

Calibre license path: 1717@calibrelic.server.com

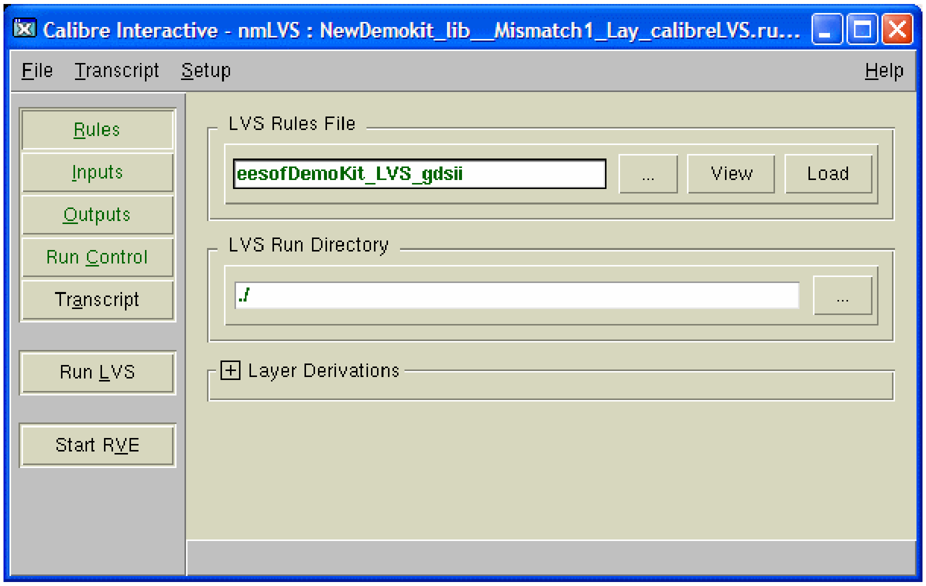
□ Use PuTTY

7. Choose **Remote** for the *Run mode* in the *Settings* dialog.
8. Enter the *Machine name* and the *Remote user name*. *Remote password* is not required.
9. Click **OK** in the *Settings* dialog to return to the *Send To Calibre Interactive* dialog.
10. In the *Schematic Design* drop down box, specify the schematic design against which you would like the Layout design to be checked for LVS errors.
11. Click **Browse** to choose the *Calibre LVS Rule file*.
12. The Job name is automatically filled up by ADS. You can edit this name.
13. Click **Send** to start the LVS flow.

> **Note**
> If you do not wish to generate the SSH Private/Public Keys, then you can click the "Use PuTTY" option for ADS running on Windows machine. In such cases, you will be prompted to enter the password every time you run a LVS job. The PuTTY installation path MUST be specified in your Windows `PATH` environment variable.

Running LVS

1. From the *Send To Calibre Interactive* dialog, click **Send**. Calibre Interactive dialog opens.



LVS Rules File: eesofDemoKit_LVS_gdsii

LVS Run Directory: ./

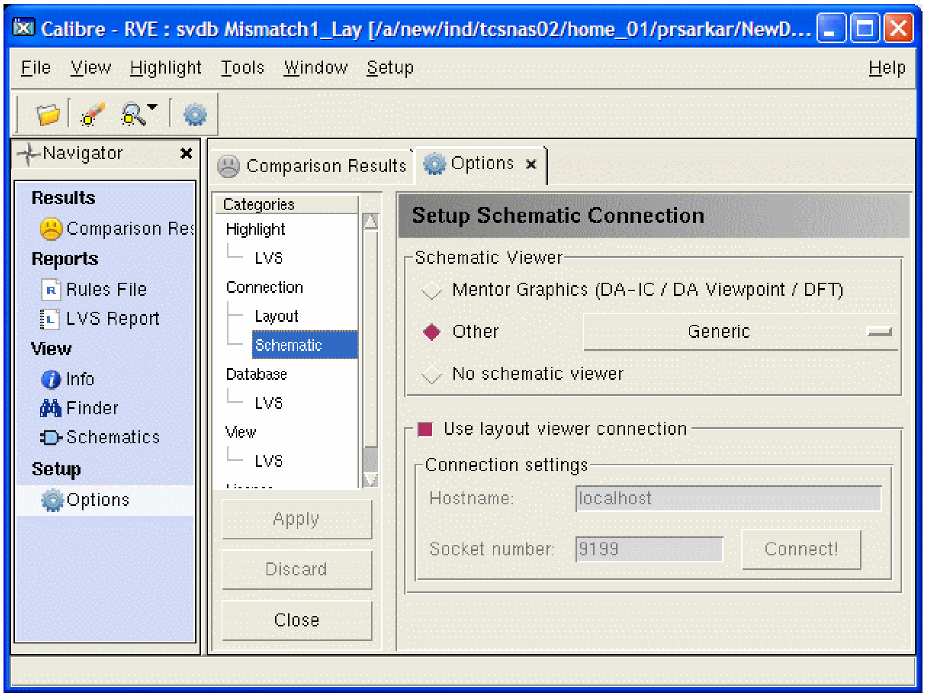
2. Depending on your network speed, it might take between 10 to 60s for Calibre Interactive to open. Wait for the Calibre GUIs to open on your desktop.
3. Click on **Run LVS**. Calibre displays the results of the LVS run and also open up RVE.

> **Important**
> If you are invoking ADS-Calibre LVS Link from a Windows machine, ensure that you have a Xserver software already running. If the XServer is not running, Calibre GUI windows will not open on your desktop and you will not be able to run LVS.

> **Note**
> If you would like certain environment variables(e.g variables that are specific to your LVS Rule deck) to be made available to Calibre, you may do so by defining them in a file *calibre_config.sh*
> This file MUST be kept inside your ADS workspace folder. If ADS finds this file while running Calibre LVS, it would first source this file on the remote UNIX machine before actually invoking Calibre.
> This provides you the flexibility to do any customization (for example: defining Rule Deck specific environment variables, copying other rules deck files etc) in the *calibre_config.sh*

Setting up Calibre RVE For Viewing LVS Errors on ADS Layout and Schematic

1. Setup Calibre RVE before viewing errors on ADS Layout and Schematic.



Setup Schematic Connection

Schematic Viewer

○ Mentor Graphics (DA-IC / DA Viewpoint / DFT)

● Other Generic

○ No schematic viewer

■ Use layout viewer connection

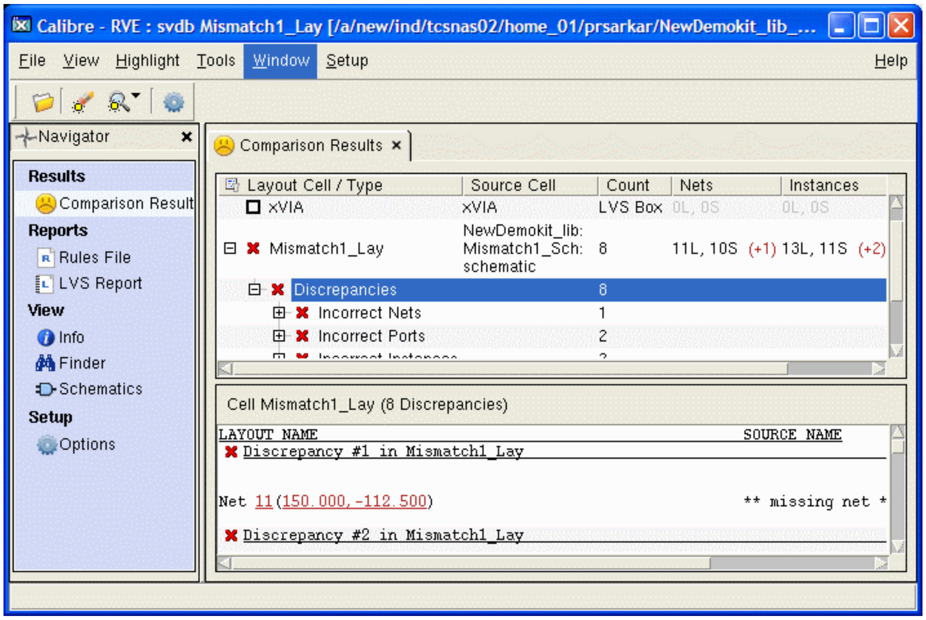
Connection settings

Hostname: localhost

Socket number: 9199

Connect!

2. In the Calibre RVE, click **Setup > Schematic Viewer**. The *Setup Schematic Viewer* dialog pops up.
3. Click on the radio button ***Other*** and set the type as ***Generic*** if not already set.
4. Click the check box titled ***Use layout viewer connection*** to enable it. Click **Apply**.
5. Click **Setup > Layout Viewer**. The *Setup Layout Viewer* dialog pops up.
6. Click on the radio button ***Other*** and set the type as ***Generic*** if not already set. Click **Apply**.
7. Position your top level layout and schematic window in a non-overlapping fashion, as it would help you to clearly see both the layout and the schematic errors.
8. You are now ready to view your LVS errors whenever you click on the error items in Calibre RVE.


Calibre - RVE : svdb Mismatch1_Lay [/a/new/ind/tcsnas02/home_01/prsarkar/NewDemokit_lib_...
File View Highlight Tools Window Setup Help
Navigator
Results
Comparison Result
Reports
Rules File
LVS Report
View
Info
Finder
Schematics
Setup
Options
Comparison Results

Layout Cell / Type	Source Cell	Count	Nets	Instances
☐ xVIA	xVIA	LVS Box	0L, 0S	0L, 0S
⊟ ✖ Mismatch1_Lay	NewDemokit_lib: Mismatch1_Sch: schematic	8	11L, 10S (+1)	13L, 11S (+2)
⊟ ✖ Discrepancies		8		
⊞ ✖ Incorrect Nets		1		
⊞ ✖ Incorrect Ports		2		

Cell Mismatch1_Lay (8 Discrepancies)

```
LAYOUT NAME                                          SOURCE NAME
✖ Discrepancy #1 in Mismatch1_Lay


Net 11(150.000,-112.500)                             ** missing net *

✖ Discrepancy #2 in Mismatch1_Lay
```

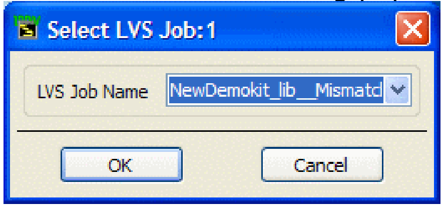
9. For details on Calibre RVE and Interactive, visit Mentor Graphics website.

Viewing LVS Errors From a Previously Run LVS Job

> **Note**
> You can view errors for only one LVS Job at a time.

You can also view the errors of a previously run LVS job, provided you have not deleted the job folder that ADS had created while sending the designs to Calibre Interactive.

1. From your Layout window, click **Tools > Calibre LVS > Start RVE Interface**
2. The *Select LVS Job* dialog pops up.



Select LVS Job:1

LVS Job Name: NewDemokit_lib__Mismatd

3. Select the LVS job name from the drop down list
4. Click **OK**.
5. ADS attempts to invoke Calibre RVE (not Calibre Interactive). Again, if you have invoked *Start RVE Interface* from a Windows machine, ensure that a XServer software is already running.
   Depending on your network speed, it might take between 10 to 60s for Calibre RVE to pop up. Wait for the Calibre GUIs to appear on your desktop. Follow the steps mentioned in
   Viewing LVS Errors on ADS Layout and Schematic.

Clearing LVS Highlighted Components

> **Note**
> When you close the layout window, the highlighted portions from only the layout window shall get cleared. The highlighted components in the schematic would remain. You can clear them using **Layout > Clear Highlighted Components** menu option.

You can clear the highlighted portions of your layout and schematic designs by using any of the following options.

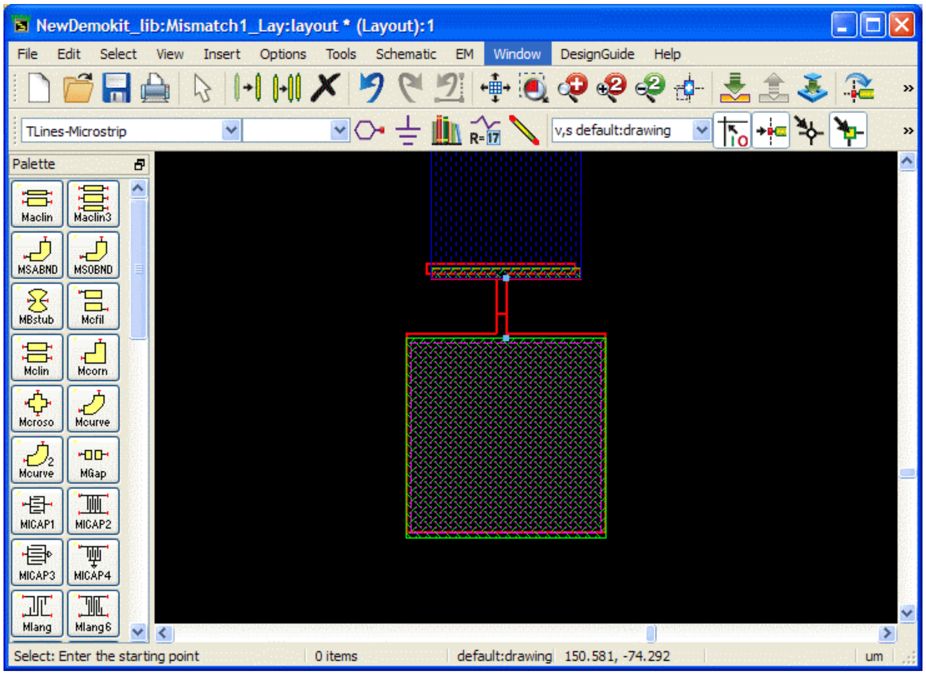
1. On the layout window, click on **Tools > Calibre LVS > Clear LVS Highlights.** This would clear highlighting from the layout and well as schematic windows
2. You can also clear the highlighted components using the menu option **View > Clear Highlights** in the Calibre RVE. This would clear highlighting from the layout and well as schematic windows

Example Snapshots

The following figures illustrates how the layout portions and the schematic components that are caught in the LVS run, are highlighted inside red bounding boxes.

A Layout Portion Being Highlighted

ADS shall highlight instances, using only a single color, the default of which is RED. ADS will use DRC layer no 100 as the default layer for highlighting Layout components. If however, layer no 100 is already being used a physical layer, the next (101, or 102 and so forth) available DRC layer shall be used for highlighting Layout components.



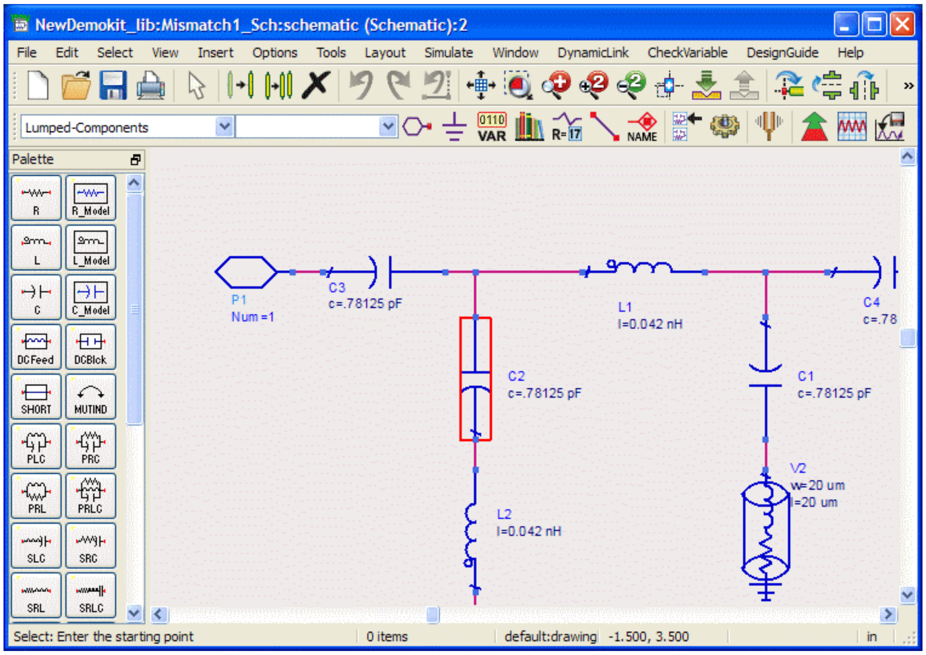
NewDemokit_lib:Mismatch1_Lay:layout * (Layout):1

TLines-Microstrip

v,s default:drawing

Select: Enter the starting point | 0 items | default:drawing 150.581, -74.292 | um

A Schematic Component Being Highlighted

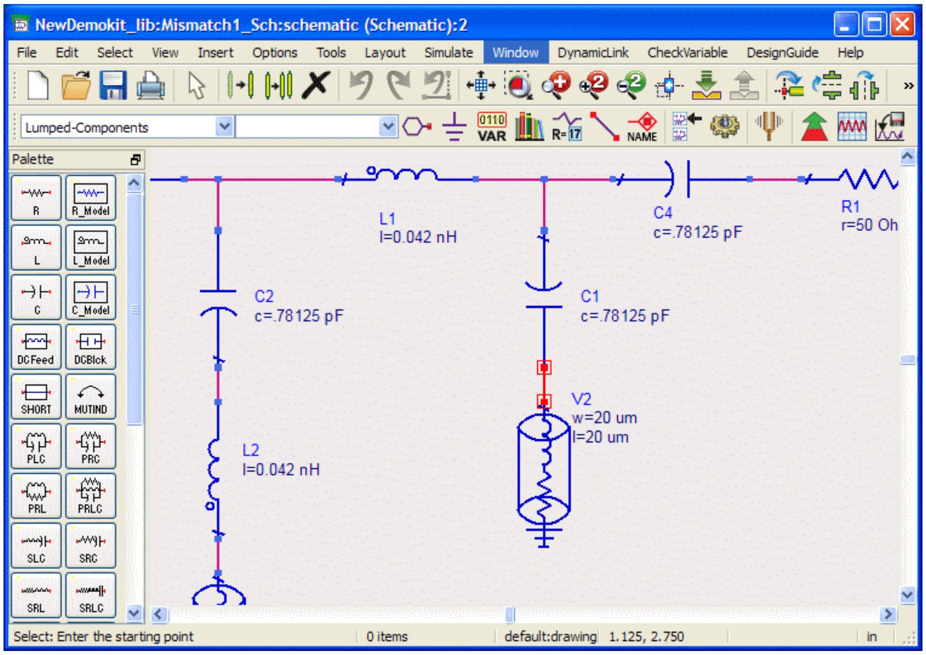


NewDemokit_lib:Mismatch1_Sch:schematic (Schematic):2

Lumped-Components

Select: Enter the starting point | 0 items | default:drawing -1.500, 3.500 | in

A Schematic Net Being Highlighted



Select: Enter the starting point | 0 items | default:drawing 1.125, 2.750 | in

Note On Schematic Designs containing EM Look-alike Components

If your schematic design contains EM Layout look-alike components, you can still run LVS on your design. The look-alike would be treated as an instance of a sub-circuit and that sub-circuit MUST have a corresponding schematic design containing the actual PDK components.

Note On Sub-circuit Instance Names

If your design contains instances of sub-circuit, then make sure that the instance names do not start with the letter 'x' or 'X'. There is an issue with Calibre RVE which prevents the schematic instances from being highlighted, when LVS is run in flat mode. As per communication received from Mentor Graphics, this issue is reported to be fixed in Calibre 2009 August Release.

Note On Case Sensitivity of Instance Names

You MUST set case sensitivity to true by specifying SOURCE CASE YES in your Calibre LVS rule file. By default, Calibre ignores case sensitivity of schematic components names whereas ADS treats them in a case sensitive manner.

Note On Probing

ADS does not support probing Net/Device/Instance from the **Source > Net Queries, Source > Device Queries, Source > Instance Queries** menu option of Calibre RVE. The same holds true for layout probing as well. However you can highlight the corresponding instances using these menu options.